Generating long streams of $1/f^{a}lpha$ noise

S. Plaszczynski

Fluctuation and Noise Letters
Vol. 0, No. 0 (2006) 000–000

Generating long streams of $1/f^\alpha$ noise

S. Plaszczynski
Laboratoire de l'Accélérateur Linéaire,
IN2P3-CNRS et Université Paris-Sud 11, Centre Scientifique d'Orsay, B.P. 34,
91898 ORSAY Cedex, France.
plaszczy@lal.in2p3.fr

Received (July 9, 2008)
Revised (revised date)
Accepted (accepted date)

We review existing methods for generating long streams of $1/f^\alpha$ noise ($0 < \alpha \leq 2$) focusing on the digital filtering of white noise. We detail the formalism to conceive an efficient $1/f^\alpha$ random number generator (white outside some bounds) in order to generate very long streams of noise without an exhaustive computer memory load. For $\alpha = 2$ it is shown why the process is equivalent to a random-walk and can be obtained simply by a first order filtering of white noise. As soon as $\alpha < 2$ the problem becomes non linear and we show why the exact digital filtering method becomes inefficient. Instead, we work out the formalism of using several $1/f^2$ filters spaced logarithmically, to approximate the spectrum at the percent level. Finally, from work on logistic maps, we give hints on how to design generators with $\alpha > 2$. The corresponding software is available from `http://planck.lal.in2p3.fr/wiki/pmwiki.php/Softs/AbsRand`.

Keywords: $1/f$ noise, simulation, random walk, digital filtering, fractals

Introduction

Many physical systems exhibits some $1/f^\alpha$ noise, *i.e.* a stochastic process with a spectral density having a power exponent $0 < \alpha \leq 2$. Among the many fascinating questions related to it [1] an apparently simple problem is how to build random number generators (RNG) with such long-range correlations. The classical method (see *e.g.* [1]) is to generate a vector of white noise (as produced by a standard RNG) with an appropriate correlation matrix. However, one needs to take the "square root" of this matrix (as with a Cholesky decomposition [2]), a method rapidly limited by the size of the requested sample. One must then turn on to more specialized techniques, which can be categorized into:

- *Digital signal processing*: one generate a vector of white noise and fast-convolve it in Fourier space with the required $1/f^\alpha$ shape (see *e.g.* [3]). One may use also a wavelet decomposition ([4]) which is quite natural due to the scaling properties of the process (see next part). With these methods samples of the order of $o(10^8)$ can be efficiently obtained. For samples with longer correlations, one is limited by the memory of the computer.

- *Fractal techniques*: in non-linear science, $1/f^\alpha$ noise is referred to as "fractional brownian noise". The peculiar shape of the spectrum (with a power exponent) implies that the process is "scale-invariant" or more precisely "self-similar" (see *e.g.* [5]), meaning that the stochastic process $X(t)$ is statistically similar (in the sense of having the same statistical moments in the infinite limit) to $\frac{X(rt)}{r^H}$ for any dilatation r, where H is the Hurst exponent related to the $1/f^\alpha$ slope by $\alpha = 2H + 1$. Some very efficient method to generate this process [6] is to shoot

[1] a large collection of references is available online:
`http://www.nslij-genetics.org/wli/1fnoise/index.html`

Gaussian numbers for both ends of an interval, take the average at the mean and add a new shot with a re-scaled amplitude, and repeat the process on each subintervals. This however requires knowing *a-priori* the limits of your sample and is also limited by the computer memory for very large samples.

In some cases, these methods are insufficient, as in some modern astrophysics experiments. For instance, this work has been perform in the framework of the simulation of the Planck ESA mission [7] which is a satellite experiment, planned to be launched in 2008, whose goal is to measure the Cosmic Microwave Background anisotropies with unprecedent accuracy. It is composed of two instruments: High Frequency (HFI) and Low Frequency (LFI), the former using low temperature bolometers (0.1K) and the latter radiometers. At the level of precision required ($\simeq 10^{-5}K$) the instruments are sensitive to $1/f^2$ noise mostly from the cryogenic parts (HFI) and $1/f^\alpha$ with $\alpha \simeq 1.7$ from the low-noise electronics (LFI). Given the sampling rate of the instruments (200 Hz) and its long duration (two years), very long streams of noise need to be generated to prepare the analyzes . The techniques described previously fail.

In the following we will show how to produce rapidly such long streams of $1/f^\alpha$ noise for an arbitrarily low frequency cutoff, without requiring massive memory load.

In the $1/f^2$ case (section 1) we will adapt some classical numerical filtering technique (Auto-Regressive Moving Average, see Appendix) that allows to build an *optimal* generator (*i.e.* only limited by the underlying standard white one). For $\alpha < 2$ (section 2) the problem becomes much more intricate, because we are moving off the linear theory and the previous method cannot be generalized. We will then adapt a method from electronics, to approximate the problem to a very good precision by a set of $1/f^2$ generators and use the previous case.

We will then give hints on how to build generators with $\alpha > 2$ (section 3) using methods based on logistic maps.

The Appendix recalls some properties of the z transform which will be our main tool.

The algorithms described hereafter are now full part of the Planck simulation programs [8] and streams of 1 year data of noise ($\simeq 6.3\ 10^9$) are generated in about 20 minutes on a single standard 2 GHz processor.

1. $1/f^2$ noise

1.1. *Why pure* $1/f^2$ *noise is random walk.*

Let us start with the pure $1/f^2$ noise case.

The idea is to start with a standard Gaussian generator and numerically filter the shots (called x_i, realizations of an $x(t)$ process) to obtain the proper power spectrum which is simply [2]:

$$S_y(\omega) = |H(j\omega)|^2 S_x(\omega) \tag{1}$$

Since x is white $S_x = 1$, so all we need is to design a filter, here of the form:

$$|H(j\omega)|^2 = \frac{1}{\omega^2} \tag{2}$$

therefore:

$$H(j\omega) = \frac{1}{j\omega} \tag{3}$$

This is simply a first order integrator and has an equivalent discrete z transform of the form (see Appendix):

$$H(z^{-1}) = \frac{1}{1 - z^{-1}} \tag{4}$$

Therefore the filtered signal is:

$$Y(z^{-1}) = H(z^{-1})X(z^{-1}) = \frac{X(z^{-1})}{1 - z^{-1}} \tag{5}$$

[2]Through the article we will use the electronic notation for $j^2 = -1$ and work in the frequency domain ω. In addition we will not mention the normalization factor σ of the input Gaussian generator that can be recovered straightforwardly by multiplying the output by $\frac{\sigma}{\sqrt{f_{\text{sample}}}}$.

which according to the summation property of the z transform (Appendix) is obtained simply from:

$$y_k = \sum_{i=0}^{k} x_i \tag{6}$$

This equation represents an MA (moving average) filter; the system is built by "remembering" at each step its previous value (random walk) and has therefore only one state variable. It has an "infinite memory".

It is slightly more convenient to write this filter in the following way:

$$(1 - z^{-1})Y(z^{-1}) = X(z^{-1}) \tag{7}$$

which accordingly to the offset properties of the z transform is obtained in the time domain through:

$$y_k - y_{k-1} = x_k \tag{8}$$

which represents a one state AR (Auto Regressive) filter.

1.2. *Whitening below some minimum frequency f_{min}*

While infinitely long streams may be theoretically interesting (does pure $1/f^\alpha$ noise has an intrinsic low frequency cutoff? Is it a stationary process?...) we are concerned here with designing a generator for real life experiments which have *finite* durations. Therefore it is assumed that there exists *some* frequency $f_{\min}$ below which the noise becomes white (which can be as long as the time-scale of the experiment).

To get a white spectrum below some frequency $\omega_0 = 2\pi f_{\min}$, we are seeking for a transfer function of the type:

$$|H(j\omega)|^2 = \frac{\omega_0^2}{\omega^2 + \omega_0^2} \tag{9}$$

$$H(j\omega) = \frac{\omega_0}{j\omega + \omega_0} \tag{10}$$

which represents a first order low-pass filter.

Its inverse Fourier transform is:

$$h(t) = \omega_0 e^{-\omega_0 t} u(t) \tag{11}$$

($u(t)$ being the unit step), which sampled at a rate $1/T$ has the z transform:

$$H(z^{-1}) = \sum_i h(iT) z^{-i} = \frac{\omega_0}{1 - e^{-\omega_0 T} z^{-1}} \tag{12}$$

Therefore:

$$Y(z^{-1})(1 - e^{-\omega_0 T} z^{-1}) = \omega_0 X(z^{-1}) \tag{13}$$

whose inverse transform is:

$$y_k = \omega_0 \; x_k + e^{-\omega_0 T} y_{k-1} \tag{14}$$

For each shot x_k one just needs to keep the previous filtered value y_{k-1} to compute the new one y_k.

1.3. *Whitening above some knee frequency f_{knee}*

In most experimental cases, noise becomes white also above some "knee" frequency f_{knee} and we incorporate it in the following.

For a filter with a white spectral behavior both below $f_{\min}$ and above f_{knee}, we symmetrize Eq.(9):

$$|H(j\omega)|^2 = \frac{\omega^2 + \omega_k^2}{\omega^2 + \omega_0^2} \tag{15}$$

$$H(j\omega) = \frac{j\omega + \omega_k}{j\omega + \omega_0} \tag{16}$$

In order to treat the numerator and denominator symmetrically, we use the bilinear transform (see Appendix):

$$w = j\frac{1-z}{1+z} \tag{17}$$

relating it to frequency via:

$$\omega \approx 2w f_{\text{sample}} \tag{18}$$

to get the z transfer function in the form:

$$H(z^{-1}) = \frac{a_0 + a_1 z^{-1}}{1 - b_1 z^{-1}} \tag{19}$$

with:

$$\begin{aligned} a_0 &= \frac{1+r_1}{1+r_0} \\ a_1 &= -\frac{1-r_1}{1+r_0} \\ b_1 &= \frac{1-r_0}{1+r_0} \end{aligned} \tag{20}$$

where the reduced frequencies are $r_1 = \pi f_{\text{knee}}/f_{\text{sample}}$ and $r_0 = \pi f_{\min}/f_{\text{sample}}$.

It is now straightforward to build the numerical filter from:

$$Y(z^{-1})(1 - b_1 z^{-1}) = (a_0 + a_1 z^{-1})X(z^{-1}) \tag{21}$$

which for any k reads:

$$y_k = a_0 x_k + a_1 x_{k-1} + b_1 y_{k-1} \tag{22}$$

This is the equation for a simple ARMA filter: for any white shot x_k one just needs to keep the previous one x_{k-1} and the previous filtered value y_{k-1}, to compute the new one.

This simple filter has been implemented in `C++` and run for 10^7 samples. Results are depicted on the upper plots of Figure 1 and show that it behaves correctly.

This method is *optimal*, in the sense that CPU-time is just limited by the speed of your white number generator. It is insensitive to whatever your maximal correlation length ($f_{\min}$) is. And obviously it does not require any memory load.

2. $1/f^\alpha$ noise

2.1. *Failure of ARMA models*

ARMA models were so successful in the $1/f^2$ noise case that we are tempted to generalize them to the $1/f^\alpha$ case. This is however far from trivial, mainly because we are moving off linear theories for which the z transform has been tailored. Let us see what happens in more details.

For a pure $1/f^\alpha$ noise case,we are seeking for a transfer function of the form:

$$|H(j\omega)|^2 = \frac{1}{\omega^\alpha} \tag{23}$$

$$H(j\omega) = \frac{1}{(j\omega)^{\alpha/2}} \tag{24}$$

Its generalized [9] inverse Fourier transform is:

$$h(t) = \frac{t^{-(1-\alpha/2)}}{\Gamma(\alpha/2)} \tag{25}$$

and its associated one-sided z transform is:

$$H(z^{-1}) = \sum_{i\geq 0} h(iT) z^{-i} = \frac{T^{-(1-\alpha/2)}}{\Gamma(\alpha/2)} \sum_i i^{-(1-\alpha/2)} z^{-i} \tag{26}$$

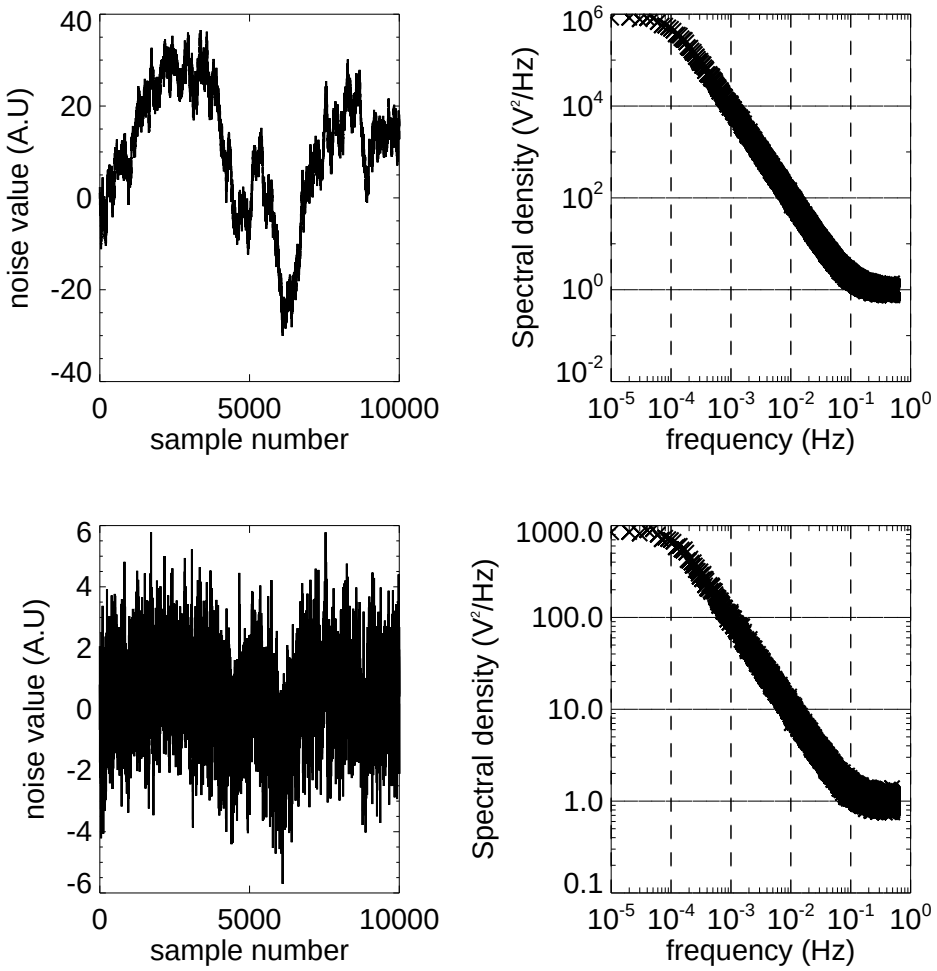

Fig 1. Results obtained with the generators. Upper plots: random walk ($\alpha = 2$). Lower plots: $1/f$ noise ($\alpha = 1$). 10^7 samples are shot in each case. A slice of the signal in real space is shown on the left, and the power spectral density on the right (estimated using a classical Welsh periodogram method [2]). The following bounds were used for whitening: $f_{\min} = 10^{-4}$ Hz, $f_{\text{knee}} = 10^{-1}$ Hz. The logarithmic scale allows to check by eye the validity of the slope.

Unfortunately the sum that appears on the r.h.s cannot be factorized, so that the MA filter $Y(z^{-1}) = H(z^{-1})X(z^{-1})$ leads to an explicit convolution:

$$y_k \propto \sum_{i=0}^{k} i^{-(1-\alpha/2)} x_{k-i} \tag{27}$$

In direct space, one can therefore produce $1/f$ noise by generating $\frac{1}{\sqrt{t}}$ type transients at random initial times [18, 19].

There are however two problems with this expression:

1. the first term $i = 0$ is infinite
2. we must keep *all* the past terms (x_{k-i})which is rapidly inefficient even with a fast convolution method.

The first problem is the reflect of the high frequency divergence of $1/f^{\alpha}$ noise [18]. Kasdin [10] proposes an elegant way to circumvent it, by using a z transfer function of the form :

$$H(z^{-1}) = \frac{1}{(1 - z^{-1})^{\alpha/2}} \tag{28}$$

(compare to (5)). It allows us to build an AR filter: $Y(z^{-1})(1-z^{-1})^{-\alpha/2} = X(z^{-1})$ by the power expansion $(1-z^{-1})^{-\alpha/2} = \sum_k \xi_k z^{-k}$:

$$\sum_{i=0}^{k} \xi_k y_{k-i} = x_k \tag{29}$$

The ξ_k tend rapidly to $k^{-(1+\alpha/2)}$ and the first term $\xi_0 = 1$ is now well defined.

This however does not solve the long range dependency which just reflects the fact that a $1/f^\alpha$ ($\alpha < 2$) spectrum has *an infinite number of state variables and that there exits no (linear) difference equations to represent it.*

One may think that this bad behavior is to due to a missing low frequency cutoff (ω_0). We can modify the transfer function (23) to see what happens:

$$|H(j\omega)|^2 = \frac{\omega_0^\alpha}{(\omega^2+\omega_0^2)^{\alpha/2}} \tag{30}$$

$$H(j\omega) = \frac{\omega_0^{\alpha/2}}{(j\omega+\omega_0)^{\alpha/2}} \tag{31}$$

then using Eq. 3.382-7 [11] , the inverse Fourier transform is:

$$h(t) = \frac{\omega_0^{\alpha/2}}{\Gamma(\alpha/2)} t^{-(1-\alpha/2)} e^{-\omega_0 t} \tag{32}$$

By comparing to the no-cutoff case Eq.(25), we see that the effect of introducing ω_0 is to attenuate the long range correlation by the exponential factor: $\xi_k \to \xi_k e^{-\omega_0 kT}$. Still, one needs to keep a few times $1/(\omega_0 T)$ past samples which is too inefficient for a low frequency cutoff.

2.2. *Recursive* $1/f^2$ *filtering*

We then turn on to an approximate method to produce efficiently long range correlated samples. An old well-known way in electronics [12] is to approximate the $1/f$ spectrum by a sum of $1/f^2$, *i.e.* relaxation processes, equally spaced on a logarithmic frequency grid, which is called the "infinite RC model" [13]. Quite surprisingly little filters are needed to produce an $1/f^\alpha$ ($0 < \alpha < 2$) spectrum at the percent level. This method is used for instance in pink audio noise generation [14], in commercial libraries [15], and even implemented on a DSP [16].

We work out here a slight variant of the RC model due to Keshner [17] which allows to incorporate whitening both below $f_{\min}$ and above f_{knee}, by filtering one single white shot.

We are seeking for a transfer function of the type:

$$|H(j\omega)|^2 = \Big|\frac{\omega^2+\omega_k^2}{\omega^2+\omega_0^2}\Big|^{\alpha/2} \tag{33}$$

The approximate transfer function is of the type:

$$|H(j\omega)|^2 \simeq \prod_{i=0}^{N_f-1} \frac{\omega^2+z_i^2}{\omega^2+p_i^2} \tag{34}$$

where N_f is the number of filters used, and p_i, z_i are the poles and zeros locations.

Since we already designed these $1/f^2$ filters in section 1.3, the only remaining point is to get their poles and zeros positions.

From Figure 2 it is clear that the poles must be located regularly on a logarithmic grid[3] :

$$\Delta p = \frac{\log\omega_k - \log\omega_0}{N_f} \tag{35}$$

$$\log p_{i+1} = \log p_i + \Delta p \tag{36}$$

[3] we use a base 10 logarithm.

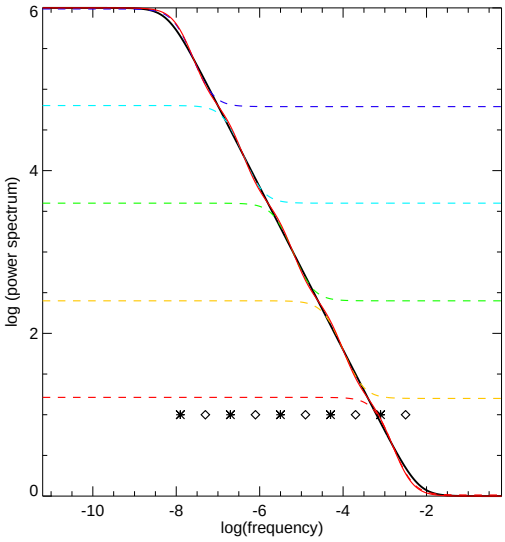

Fig 2. A band-limited $1/f$ spectrum (full line) can be approximated by a set of $1/f^2$ spectra properly located (dashed lines) . The crosses and diamonds represent respectively the position of the poles and zeros in Laplace space. Here we use only one filter per decade; the approximation is already good.

in order to have a symmetric errors, the zeros must be placed at [20]:

$$\log z_i = \log p_i + \frac{\alpha}{2}\Delta p \tag{37}$$

and for continuity, from geometrical considerations, the first pole must be put at:

$$\log p_0 = \log \omega_0 + \frac{1}{2}(1 - \frac{\alpha}{2})\Delta p \tag{38}$$

How many $1/f^2$ filters do we need? While in principle, as few as one per decade is enough to approximate the slope to better than 5% [17], since we also wish here to reproduce neatly the $f_{\min}$ and f_{knee} shoulders, we used 1.5 filters per decade, which leads to an approximation at the percent level for any $0 < \alpha < 2$ value throughout the spectrum [4].

We implemented this algorithm, using the previous code for $1/f^2$ noise: a Gaussian white number is shot and passed recursively through the set of $1/f^2$ filters: the coefficients of Eq (20) are automatically recomputed by the class for a given pole/zero and the output of each filter is the input to the next one.

We illustrate on the lower plots of Figure 1 results obtained by generating 10^7 samples of an $1/f$ noise. It required using only six $1/f^2$ filters and the increase in CPU-time with respect to a pure white Gaussian shot, is only of 25%: the limitation is mainly due to the white time shot, not the filtering stages.

The strong feature of this method is that the number of filters used increases only logarithmically down to $f_{\min}$. The drawback is that the generator has to be "warned up" to avoid the transient response, on a timescale of the order of the slowest RC filter (approximately $1/f_{\min}$).

3. $\alpha > 2$?

While rarely encountered in experimental data, the curious reader may wonder how to generate an $1/f^\alpha$ noise with a slope $\alpha > 2$. From geometric arguments, the methods presented before fails. An interesting approach is to use the work on logistic maps.

Consider the straightforward (non-linear) recurrence [21]:

$$x_{k+1} = x_k + x_k^2 \ (\texttt{mod}\ 1) \tag{39}$$

[4]The worst case being the $1/f$ spectrum.

where $0 < x_0 < 1$.

This is a pure $1/f$ noise generator (actually $1/f(\log f)^2$)!

By changing the power exponent, group theory computations show [22] that any factor $\alpha > 2$ can be reached. By adding a "shift from tangency" ϵ in the recurrence:

$$x_{k+1} = (1+\epsilon)x_k + (1-\epsilon)x_k^2 \ (\texttt{mod}\ 1) \tag{40}$$

one introduces a low frequency cutoff $f_{\min}$. Whitening above f_{knee} can be obtained by shooting a properly weighted extra white shot.

This chaotic process is however far from Gaussian (it is an "intermittent" one, representing long phases of low values, followed by sudden "bursts" of high ones) but, from the Central Limit Theorem, Gaussianity can be recovered by summing up several such independent generators, a method particularly efficient on parallel computers, but beyond the scope of this article.

For completeness, note an interesting connection between this approach and a more empirical one consisting in re-scaling some Brownian motion (but in which case only $0 < \alpha < 2$ range can be reached) [23].

Conclusion

We have worked out the formalism of filtering white noise efficiently to produce "infinite" streams of band-limited $1/f^\alpha$ noise.

- For $1/f^2$ noise the method is optimal, *i.e.* is equivalent to generating standard Gaussian white noise.
- For $0 < \alpha < 2$ we use an excellent approximation which only requires about one and half$1/f^2$ filters/decade, allowing the very fast generation of $1/f^\alpha$ noise over an arbitrarily large frequency range.

The corresponding (C++) software can be downloaded from:
`http://planck.lal.in2p3.fr/wiki/pmwiki.php/Softs/AbsRand`

We have focussed mainly on this method because it is lightweight and very efficient for very long time streams. For more general applications, before choosing a dedicated algorithm ask yourself the following questions:

1. Do I want to generate a Gaussian noise?
2. What α range do I need?
3. Do I have a low frequency cutoff $f_{\min}$ below which noise is white? Note that the upper f_{knee} frequency can always be added by shooting an extra white noise (properly weighted).
4. On how many decades do I need the logarithmic slope?
5. What is my hardware support (mainly memory load)?

Table 1 then gives you some hints about which algorithm you can use.

Method	Gaussian?	α range	$f_{\min}$	CPU/load	ref.
FFT/FWT	yes	any	yes	moderate	*e.g.* [3]/ [4]
Random pulses	yes	$0 < \alpha \leq 2$	yes	high	[18], [19]
ARMA filtering	yes	$0 < \alpha \leq 2$	yes	high	[10], section 2.1
Sum of $1/f^2$ filters	yes	$0 < \alpha \leq 2$	yes	low	[15], [14], [16],section 2.2
Random midpoint displacement	yes	$1 \leq \alpha \leq 3$	no	low	[6], [5]
Brownian scaled motion	no	$0 < \alpha \leq 2$	no	low	[23]
Logistic maps	no	any	yes	low	[21], [22]

Table 1. Summary of today's existing methods to generate discrete $1/f^\alpha$ noise. The column "$f_{\min}$ " refers to the (possible) existence of a frequency limit in the method, beyond which noise is white. "CPU/load" is indicative (it should be considered as relative to the different methods) and only significant for very long correlation lengths.

Appendix

In this appendix we recall a few results of the theory of automatism that are relevant to this analysis. The reader is referred to any book on linear theories (as [24]) for more details.

The z transform is a tool similar to the Fourier transform, but for the discrete domain. To a causal series of values $\{x_{i=0,1,..}\}$ we associate its z transform by:

$$X(z^{-1}) = x_0 + x_1 z^{-1} + x_2 z^{-2} + ... \tag{.1}$$

where z is a complex number.

The following properties can then be demonstrated:

$$\begin{array}{rc} \text{linearity} & \lambda_1\{x_i\} + \lambda_2\{y_i\} \xrightarrow{z} \lambda_1 X(z^{-1}) + \lambda_2 Y(z^{-1}) \\ \text{offset} & \{x_{i-k}\} \xrightarrow{z} z^{-k} X(z^{-1}) \\ \text{summation} & \{\sum_{k=1}^{i} x_k\} \xrightarrow{z} \dfrac{X(z^{-1})}{1-z^{-1}} \\ \text{convolution} & \{\sum_{k=0}^{i} x_k y_{i-k}\} \xrightarrow{z} X(z^{-1})Y(z^{-1}) \end{array} \tag{.2}$$

A time signal $x(t)$ sampled at a period T is described as:

$$x^*(t) = \sum x(iT)\delta(t - iT) \tag{.3}$$

and we can associate to the samples $x_i = x(iT)$ a z transform. Some usual transforms are:

$$\delta(t) \xrightarrow{z} 1 \tag{.4}$$

$$u(t) \xrightarrow{z} \sum_{i=0}^{\infty} z^{-i} = \frac{1}{1-z^{-1}} \tag{.5}$$

$$e^{\omega_0 t}u(t) \xrightarrow{z} \sum_{i=0}^{\infty} (e^{\omega_0 T}z)^{-i} = \frac{1}{1-e^{\omega_0 T}z^{-1}} \tag{.6}$$

An interesting property is related to energy conservation. The power of the *sampled* signal is related to its z transform:

$$|X^*(j\omega)|^2 = X(z^{-1})X(z)_{|z=e^{j\omega T}} \tag{.7}$$

The link between $|H^*(j\omega)|$ and $|H(j\omega)|$ being not convenient, it is sometimes interesting to use the w bilinear (or Tustin) transform which is defined as

$$w = j\frac{1-z}{1+z} \tag{.8}$$

It can be shown that it is related to the frequency of the signal ω by

$$w = \tan(\omega T/2) \tag{.9}$$

In the limit of fast sampling $\omega T \to 0$ (which is most often the case since we work below Nyquist frequency) the w transform can be identified with ω, giving a convenient way to determine the z transform coefficients directly from the signal transfer function.

Finally, from the properties (.2), a rational polynomial z transfer function of the form

$$H(z^{-1}) = \frac{Y(z^{-1})}{X(z^{-1})} = \frac{a_0 + a_1 z^{-1} + ...}{1 + b_1 z^{-1} + ...} \tag{.10}$$

leads to the following filtering in the discrete time domain:

$$y_k + b_1 y_{k-1} + ... = a_0 x_k + a_1 x_{k-1} + ... \tag{.11}$$

when only a_0 is different from 0, it is an called Auto regressive (**AR**) filter (also know as Infinite Impulse Response). If all the b_i's are null, this is a Moving Average (**MA**) one (also know as Finite Impulse Response). In the general case this is called an **ARMA** filter.

References

[1] Ripley, B. D. ,"Stochastic Simulation.", Wiley, (1987)

[2] Press, W. H. *et al.*,, "Numerical recipes in C++", Cambridge University Press.

[3] V. J. Martinez and E. Saar, "Statistics of the galaxy distribution", Chapman&Hall/CRC, 2002.

[4] G. Wornell, "Wavelet-based representations for the 1/f family of fractal processes", *Proc. IEEE*, vol. **81**, No. 10 (1993).

[5] H. O Peitgen *et al.*,, "Chaos and fractals", Springer-Verlag (1992)

[6] A. Fournier, D. Fussel and L. Carpenter, "Computer rendering of stochastic models", *Comm of the ACM* **25** (1982).

[7] Planck web site: `http://www.rssd.esa.int/index.php?project=PLANCK`

[8] M. Reinecke *et al.*,, "A simulation pipeline for the Planck mission", *preprint* astro-ph/0508522, available from `http://arxiv.org/`.

[9] G. A. Campbell and R. M. Foster, "Fourier Integrals for Practical Applications", *D. Van Nostrand Co., Inc., New York* (1961)

[10] N. Kasdin, "Discrete simulation of colored noise and stochastic-processes and $1/f^{\alpha}$ power-law noise generation", *Proc. IEEE*, vol. **83** (1995).

[11] I.S. Gradshteyn and M. Ryzhik, "Table of integrals, series and products", *New York: Academic Press* (1965)

[12] J. Bernamont, "Fluctuations in the resistance of thin films", *Proc. Phys. Soc.* **49** (1937),138.

[13] A. Van der Ziel, "On the noise spectra of semi-conductor noise and of flicker effect", *Physica*, **16** (1950).
J.A Barnes and D.W. Allan, "A statistical model of flicker noise", *Proc. IEEE*,**54** (1966).

[14] Robin Whittle "DSP generation of Pink (1/f) Noise" `http://www.firstpr.com.au/dsp/pink-noise/`

[15] National Instruments, "Pink Noise Generator", `http://zone.ni.com/devzone/cda/epd/p/id/1023`

[16] R. Mingesz, Z. Gingl, P. Makra, "DSP based $1/f^{\alpha}$ noise generator". *Fluctuation and Noise Letters*, **vol. 4**, (2004).

[17] M. Keshner, "1/f noise", *Proc. IEEE*, vol. **70** (1982).

[18] V. Radeka, "1/f noise in physical measurements", *IEEE Trans. Nucl. Sci.* **16**, 17 (1969).

[19] A. Ambrozy and L.B. Kiss, "1/f noise generator", *Proc. 8th Intern. Conf. on Noise in Physical Systems*, eds. A. D'Amico and P. Mazzetti, North-Holland (1986) p. 445

[20] R. Saletti, " A comparison between two methods to generate $1/f^{\gamma}$ noise", *Proc.IEEE*,vol. **74** (1986).

[21] P. Manneville, "Intermittency, self-similarity and 1/f spectrum in dissipative dynamical systems", *J. Phys* **41** (1080).

[22] A. Ben-Mizrachi *et al.*,, "Real and Apparent Divergences in Low-Frequency Spectra of Nonlinear Dynamical Systems", *Phys. Rev.* **A31** (1985).

[23] Z. Gingl, L.B. Kiss and R. Vajtai, "$1/f^{k}$ noise generated by scaled Brownian motion", *Solid State Comm.* **71** (1989) 765

[24] R. Schwartz and B. Friedland, "Linear systems", McGraw-Hill (1965).
A.V. Oppenheim and R.W. Schafer. "Digital Signal Processing", Prentice-Hall, Inc.: Englewood Cliffs, NJ (1975).